The main idea behind cloud computing is not a new one. John McCarthy in the 1960s already envisioned that computing facilities will be provided to the general public like a utility [39]. The term “cloud” has also been used in various contexts such as describing large ATM networks in the 1990s. However, it was after Google’s CEO Eric Schmidt used the word to describe the business model of providing services across the Internet in 2006, that the term really started to gain popularity.

**NIST definition of cloud computing** *Cloud computing is a model for enabling convenient*, *on-demand network access to a shared pool of configurable computing resources* (*e*.*g*., *networks*, *servers*, *storage*, *applications*, *and services*) *that can be rapidly provisioned and released with minimal management effort or service provider interaction*.

***Highly scalable***: Infrastructure providers pool large amount of resources from data centers and make them easily accessible. A service provider can easily expand its service to large scales in order to handle rapid increase in service demands (e.g., flash-crowd effect). This model is sometimes called **surge computing**.

***No up-front investment***: Cloud computing uses a pay-asyou-go pricing model. A service provider does not need to invest in the infrastructure to start gaining benefit from cloud computing. It simply rents resources from the cloud according to its own needs and pay for the usage.

***Lowering operating cost***: Resources in a cloud environment can be rapidly allocated and de-allocated on demand. Hence, a service provider no longer needs to provision capacities according to the peak load. This provides huge savings since resources can be released to save on operating costs when service demand is low.

***Easy access***: Services hosted in the cloud are generally web-based. Therefore, they are easily accessible through a variety of devices with Internet connections. These devices not only include desktop and laptop computers, but also cell phones and PDAs.

***Reducing business risks and maintenance expenses***: By outsourcing the service infrastructure to the clouds, a service provider shifts its business risks (such as hardware failures) to infrastructure providers, who often have better expertise and are better equipped for managing these risks. In addition, a service provider can cut down the hardware maintenance and the staff training costs.

**Related Technologies:**

***Grid Computing***: Grid computing is a distributed computing paradigm that coordinates networked resources to

achieve a common computational objective. The development of Grid computing was originally driven by scientific applications which are usually computation-intensive. Cloud computing is similar to Grid computing in that it also employs distributed resources to achieve application-level objectives. However, cloud computing takes one step further by leveraging virtualization technologies at multiple levels (hardware and application platform) to realize resource sharing and dynamic resource provisioning.

***Utility Computing***: Utility computing represents the model of providing resources on-demand and charging customers based on usage rather than a flat rate. Cloud computing can be perceived as a realization of utility computing. It adopts a utility-based pricing scheme entirely for economic reasons. With on-demand resource provisioning and utility-based pricing, service providers can truly maximize resource utilization and minimize their operating costs.

***Virtualization***: Virtualization is a technology that abstracts away the details of physical hardware and provides

virtualized resources for high-level applications. A virtualized server is commonly called a virtual machine (VM). Virtualization forms the foundation of cloud computing, as it

provides the capability of pooling computing resources from clusters of servers and dynamically assigning or reassigning virtual resources to applications on-demand.

***Autonomic Computing***: Originally coined by IBM in 2001, autonomic computing aims at building computing systems capable of self-management, i.e. reacting to internal and external observations without human intervention. The goal of autonomic computing is to overcome the management complexity of today’s computer systems. Although cloud computing exhibits certain autonomic features such as automatic resource provisioning, its objective is to lower the resource cost rather than to reduce system complexity.



***Distributed file system over clouds***

**Google File System** (**GFS**) [19] is a proprietary distributed file system developed by Google and specially designed to provide efficient, reliable access to data using large clusters of commodity servers. Files are divided into chunks of 64 megabytes, and are usually appended to or read and only extremely rarely overwritten or shrunk. Compared with traditional file systems, GFS is designed and optimized to run on data centers to provide extremely high data throughputs, low latency and survive individual server failures. Inspired by GFS, the open source **Hadoop Distributed File System** (**HDFS**) [24] stores large files across multiple machines. It achieves reliability by replicating the data across multiple servers. Similarly to GFS, data is stored on multiple geo-diverse nodes. The file system is built from a cluster of data nodes, each of which serves blocks of data

over the network using a block protocol specific to HDFS. Data is also provided over HTTP, allowing access to all content from a web browser or other types of clients. Data nodes can talk to each other to rebalance data distribution, to move copies around, and to keep the replication of data high.

***Distributed application framework over clouds***

HTTP-based applications usually conform to some web application framework such as Java EE. In modern data center environments, clusters of servers are also used for computation and data-intensive jobs such as financial trend analysis, or film animation.

**MapReduce** [16] is a software framework introduced by Google to support distributed computing on large data sets on clusters of computers. MapReduce consists of one Master,

to which client applications submit MapReduce jobs. The Master pushes work out to available task nodes in the

data center, striving to keep the tasks as close to the data as possible. The Master knows which node contains the

data, and which other hosts are nearby. If the task cannot be hosted on the node where the data is stored, priority is given to nodes in the same rack. In this way, network traffic on the main backbone is reduced, which also helps to improve throughput, as the backbone is usually the bottleneck. If a task fails or times out, it is rescheduled. If the Master fails, all ongoing tasks are lost. The Master records what it is up to in the filesystem. When it starts up, it looks for any such data, so that it can restart work from where it left off. The open source **Hadoop MapReduce project** [25] is inspired by Google’s work. Currently, many organizations are using Hadoop MapReduce to run large data-intensive computations.



**Research challenges**

**Automated service provisioning:** One of the key features of cloud computing is the capability of acquiring and releasing resources on-demand. The objective of a service provider in this case is to allocate and de-allocate resources from the cloud to satisfy its service level objectives (SLOs), while minimizing its operational cost. However, it is not obvious how a service provider can achieve this objective. In particular, it is not easy to determine how to map SLOs such as QoS requirements to low-level resource requirement such as CPU and memory requirements. Furthermore, to achieve high agility and respond to rapid demand fluctuations such as in flash crowd effect, the resource provisioning decisions must be made online. Automated service provisioning is not a new problem. Dynamic resource provisioning for Internet applications has been studied extensively in the past [47, 57]. These approaches typically involve: (1) Constructing an application performance model that predicts the number of application instances required to handle demand at each particular level, in order to satisfy QoS requirements; (2) Periodically predicting future demand and determining resource requirements

using the performance model; and (3) Automatically allocating resources using the predicted resource requirements. Application performance model can be constructed using various techniques, including Queuing theory [47], Control theory [28] and Statistical Machine Learning [7]. Additionally, there is a distinction between proactive and reactive resource control. The proactive approach uses predicted demand to periodically allocate resources before they are needed. The reactive approach reacts to immediate demand fluctuations before periodic demand prediction is available. Both approaches are important and necessary for effective resource control in dynamic operating environments.

**Virtual machine migration:** Virtualization can provide significant benefits in cloud computing by enabling virtual machine migration to balance load across the data center. In addition, virtual machine migration enables robust and highly responsive provisioning in data centers. Virtual machine migration has evolved from process

migration techniques [37]. More recently, Xen [55] and VMWare [52] have implemented “live” migration of VMs that involves extremely short downtimes ranging from tens of milliseconds to a second. Clark et al. [13] pointed out that migrating an entire OS and all of its applications as one unit allows to avoid many of the difficulties faced by process-level migration approaches, and analyzed the benefits of live migration of VMs. The major benefits of VM migration is to avoid hotspots; however, this is not straightforward. Currently, detecting workload hotspots and initiating a migration lacks the agility to respond to sudden workload changes. Moreover, the inmemory state should be transferred consistently and efficiently, with integrated consideration of resources for applications and physical servers.

**Server consolidation:** Server consolidation is an effective approach to maximize resource utilization while minimizing energy consumption in a cloud computing environment. Live VMmigration technology is often used to consolidate VMs residing on multiple under-utilized servers onto a single server, so that the remaining servers can be set to an energy-saving state. The problem of optimally consolidating servers in a data center is often formulated as a variant of the vector bin-packing problem [11], which is an NP-hard optimization problem. Various heuristics have been proposed for this problem [33, 46]. Additionally, dependencies among VMs, such as communication requirements, have also been considered recently[34]. However, server consolidation activities should not hurt application performance. It is known that the resource usage (also known as the **footprint** [45]) of individual VMs may vary over time [54]. For server resources that are shared among VMs, such as bandwidth, memory cache and disk I/O, maximally consolidating a server may result in resource

congestion when a VM changes its footprint on the server [38]. Hence, it is sometimes important to observe

the fluctuations of VM footprints and use this information for effective server consolidation. Finally, the system must quickly react to resource congestions when they occur [54].

**Energy management:** Improving energy efficiency is another major issue in cloud computing. It has been estimated that the cost of powering and cooling accounts for 53% of the total operational expenditure of data centers [26]. In 2006, data centers in the US consumed more than 1.5% of the total energy generated in that year, and the percentage is projected to grow 18% annually[33]. Hence infrastructure providers are under enormous pressure to reduce energy consumption. The goal is not only to cut down energy cost in data centers, but also to meet government regulations and environmental standards. Designing energy-efficient data centers has recently received considerable attention. This problem can be approached from several directions. For example, energy-efficient hardware architecture that enables slowing down CPU speeds and turning off partial hardware components[8] has become commonplace. Energy-aware job scheduling [50] and server consolidation [46] are two other ways to reduce power consumption by turning off unused machines. Recent research has also begun to study energy-efficient network protocols and infrastructures [27]. A key challenge in all the above methods is to achieve a good trade-off between energy savings and application performance. In this respect,

few researchers have recently started to investigate coordinated solutions for performance and power management in a dynamic cloud environment [32].

**Traffic management and analysis:** Analysis of data traffic is important for today’s data centers. For example, many web applications rely on analysis of traffic data to optimize customer experiences. Network operators also need to know how traffic flows through the network in order to make many of the management and planning decisions. However, there are several challenges for existing traffic measurement and analysis methods in Internet Service Providers (ISPs) networks and enterprise to extend to data centers. Firstly, the density of links is much higher than that in ISPs or enterprise networks, which makes the worst-case scenario for existing methods. Secondly, most existing methods can compute traffic matrices between a few hundreds end hosts, but even a modular data center can have several thousand servers. Finally, existing methods usually assume some flow patterns that are reasonable in Internet and enterprises networks, but the applications deployed on data centers, such as MapReduce jobs, significantly change the traffic pattern. Further, there is tighter coupling in application’s use of network, computing, and storage resources, than what is seen in other settings. Currently, there is not much work on measurement and analysis of data center traffic. Greenberg et al. [21] report data center traffic characteristics on flow sizes and concurrent flows, and use these to guide network infrastructure design. Benson et al. [16] perform a complementary study of traffic at the edges of a data center by examining SNMP traces from routers.

**Data security:** Data security is another important research topic in cloud computing. Since service providers typically do not have access to the physical security system of data centers, they must rely on the infrastructure provider to achieve full data security. Even for a virtual private cloud, the service provider can only specify the security setting remotely, without knowing whether it is fully implemented. The infrastructure provider, in this context, must achieve the following objectives: (1) *confidentiality*, for secure data access and transfer, and (2) *auditability*, for attesting whether security setting of applications has been tampered or not. Confidentiality is usually achieved using cryptographic protocols, whereas auditability can be achieved using remote attestation techniques. Remote attestation typically requires a trusted platform module (TPM) to generate non-forgeable

system summary (i.e. system state encrypted using TPM’s private key) as the proof of system security. However, in a virtualized environment like the clouds, VMs can dynamically migrate from one location to another, hence directly using remote attestation is not sufficient. In this case, it is critical to build trust mechanisms at every architectural layer of the cloud. Firstly, the hardware layer must be trusted using hardware TPM. Secondly, the virtualization platform must be trusted using secure virtual machine monitors [43]. VM migration should only be allowed if both source and destination servers are trusted. Recent work has been devoted to designing efficient protocols for trust establishment and management [31, 43].

**Software frameworks:** Cloud computing provides a compelling platform for hosting large-scale data-intensive applications. Typically, these applications leverage MapReduce frameworks such as Hadoop for scalable and fault-tolerant data processing. Recent work has shown that the performance and resource consumption of a MapReduce job is highly dependent on the type of the application [29, 42, 56]. For instance, Hadoop

tasks such as sort is I/O intensive, whereas grep requires significant CPU resources. Furthermore, the VM allocated to each Hadoop node may have heterogeneous characteristics. For example, the bandwidth available to a VM is dependent on other VMs collocated on the same server. Hence, it is possible to optimize the performance and cost of a MapReduce application by carefully selecting its configuration parameter values [29] and designing more efficient scheduling algorithms [42, 56]. By mitigating the bottleneck resources, execution time of applications can be significantly improved. The key challenges include performance modeling of Hadoop jobs (either online or offline), and adaptive scheduling in dynamic conditions. Another related approach argues for making MapReduce frameworks energy-aware [50]. The essential idea of this approach is to turn Hadoop node into sleep mode when it has finished its job while waiting for new assignments. To do so, both Hadoop and HDFS must be made energy-aware. Furthermore, there is often a trade-off between performance and energy-awareness. Depending on the objective, finding a desirable trade-off point is still an unexplored research topic.

**Storage technologies and data management:** Software frameworks such as MapReduce and its various implementations such as Hadoop and Dryad are designed for distributed processing of data-intensive tasks. As mentioned previously, these frameworks typically operate on Internet-scale file systems such as GFS and HDFS. These file systems are different from traditional distributed file systems in their storage structure, access pattern and application programming interface. In particular, they do not implement the standard POSIX interface, and therefore introduce compatibility issues with legacy file systems and applications. Several research efforts have studied this problem [4, 40]. For instance, the work in [4] proposed a method for supporting the MapReduce framework using cluster file systems such as IBM’s GPFS. Patil et al. [40] proposed new API primitives for scalable and concurrent data access.

**Novel cloud architectures:** Currently, most of the commercial clouds are implemented in large data centers and operated in a centralized fashion. Although this design achieves economy-of-scale and high manageability, it also comes with its limitations such high energy expense and high initial investment for constructing data centers. Recent work [12, 48] suggests that smallsize data centers can be more advantageous than big data centers in many cases: a small data center does not consume so much power, hence it does not require a powerful and yet expensive cooling system; small data centers are cheaper to build and better geographically distributed than large data centers. Geo-diversity is often desirable for response time-critical services such as content delivery and interactive gaming. For example, Valancius et al. [48] studied the feasibility of hosting video-streaming services using application gateways (a.k.a. nano-data centers). Another related research trend is on using voluntary resources (i.e. resources donated by end-users) for hosting cloud applications [9]. Clouds built using voluntary resources, or a mixture of voluntary and dedicated resources are much cheaper to operate and more suitable for non-profit applications such as scientific computing. However, this architecture also imposes challenges such managing heterogeneous resources and frequent churn events. Also, devising incentive schemes for such architectures is an open research problem.

* Virtualization further enhances flexibility because it abstracts the hardware to the point where software stacks can be deployed and redeployed without being tied to a specific physical server. Virtualization enables a dynamic datacenter where servers provide a pool of resources that are harnessed as needed, and where the relationship of applications to compute, storage, and network resources changes dynamically in order to meet both workload and business demands. With application deployment decoupled from server deployment, applications can be deployed and scaled rapidly, without having to first procure physical servers. Virtual machines have become the prevalent abstraction — and unit of deployment — because they are the least-common denominator interface between service providers and developers. Using virtual machines as deployment objects is sufficient for 80 percent of usage, and it helps to satisfy the need to rapidly deploy and scale applications.

To the ***technological***aspects belong in particular issues related to (1) **scale and elastic scalability**, which is not only currently restricted to horizontal scale out, but also inefficient as it tends to resource over usage due to limited scale down capabilities and full replication of instances rather than only of essential segments. (2) **Trust, security and privacy** always pose issues in any internet provided service, but due to the specific nature of clouds, additional aspects related e.g. to multitenancy arise and control over data location etc. arise. What is more, clouds simplify malicious use of resources, e.g. for hacking purposes, but also for sensitive calculations (such as weapon design) etc. (3) **Handling data** in clouds is still complicated - in particular as data size and diversity grows, pure replication is no viable approach, leading to consistency and efficiency issues. Also, the lacking control over data location and missing provenance poses security and legalistic issues. (4) **Programming models** are currently not aligned to highly scalable applications and thus do not exploit the capabilities of clouds, whilst they should also simplify development. Along the same line, developers, providers and users should be able to control and restrict distribution and scaling behaviour. This relates to (5) **systems development and management** which is currently still executed mostly manually, thus contributing to substantial efficiency and bottleneck issues.

On the other hand, ***non-technological***issues play a major role in realizing these technological aspects and in ensuring viability of the infrastructures in the first instance. To these belong in particular (1) **economic** aspects which cover knowledge about when, why, how to use which cloud system how this impacts on the original infrastructure (provider) –long-term experience is lacking in all these areas; and (2) **legalistic** issues which come as a consequence from the dynamic (location) handling of the clouds, their scalability and the partially unclear legislative issues in the internet. This covers in particular issues related to intellectual property rights and data protection. In addition, (3) aspects related to **green IT** need to be elaborated further, as the cloud offers principally “green capabilities” by reducing unnecessary power consumption, given that good scaling behaviour and good economic models are in place.